

SN3262 – Network Administration, Management & Security

Network Management Protocols (SNMP)

Now up to version 3.

Reading:

SNMPv1 RFC 1155, 1157, 1212, 1213

SNMPv3 RFC 2271-2275

Some useful URLs that you should look at.

http://www.unix.org.ua/orelly/perl/sysadmin/appe_01.htm

[http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l21.d/index.html#\(1\)](http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l21.d/index.html#(1))

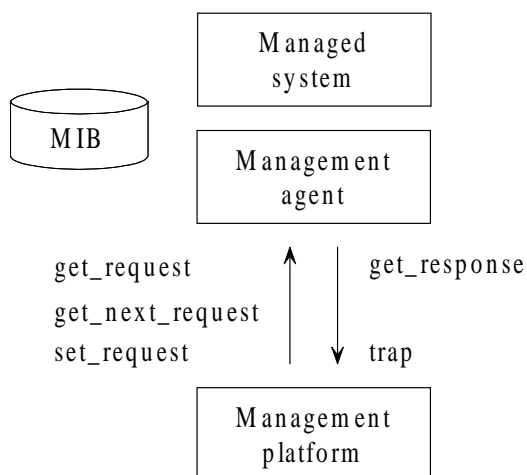
[http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l22.d/index.html#\(1\)](http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l22.d/index.html#(1))

[http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l23.d/index.html#\(1\)](http://ironbark.bendigo.latrobe.edu.au/subjects/CN/2008/lectures/l23.d/index.html#(1))

<http://www.et.put.poznan.pl/snmp/main/mainmenu.html>

All above [Accessed 22/10/08]

Basic Operation



- SNMP agent responds to queries from SNMP station

- Queries about information within the MIB
- Agent located at managed system
- Station = management application / platform

- Agent-station communication via SNMP data units

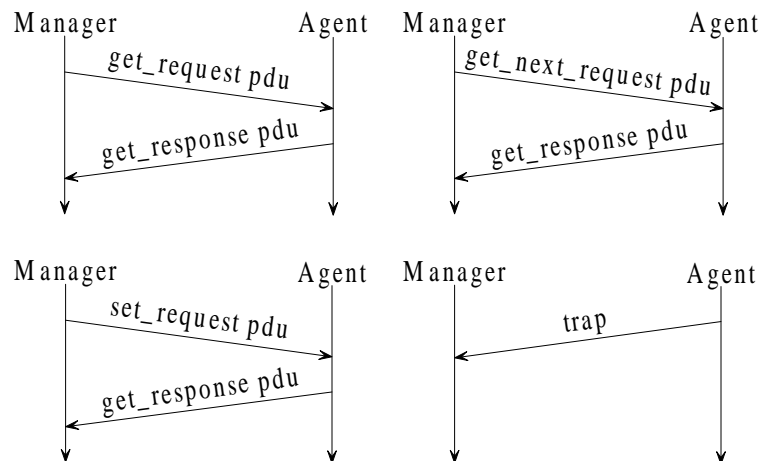
- carried inside UDP packets (UDP — unreliable, connectionless, low overhead)

Five messages types in SNMPv1

- get_request
 - allows station to query agent for specific MIB data
- get_response
 - agent supplies that data or an error indication (why the request cannot be processed)
- get_next_request
 - queries the next data item to that specified

- `set_request`
 - allows station to change data values
 - * agent replies with `get_response` containing new values (if successful) or an error indication (why the change cannot be made)
- `trap`
 - allows agent to notify station of some event
 - * no reply from management platform

PDU Exchanges



PDU Formats

Ver	Comm	SNMP PDU
-----	------	----------

SNMP message

PDU type	request-id	0	0	variablebindings
----------	------------	---	---	------------------

GetRequest, GetNextRequest and SetRequest PDUs

PDU type	request-id	error-status	error-index	variablebindings
----------	------------	--------------	-------------	------------------

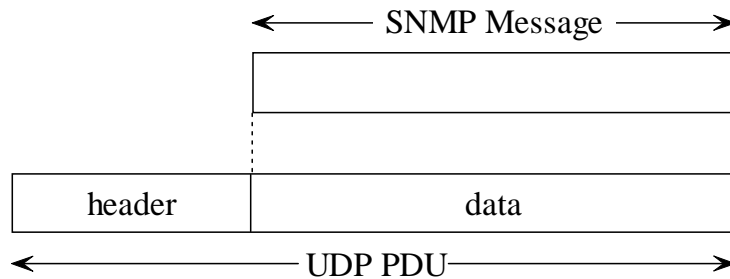
GetResponse PDU

PDU type	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variablebindings
----------	------------	------------	--------------	---------------	------------	------------------

Trap PDU

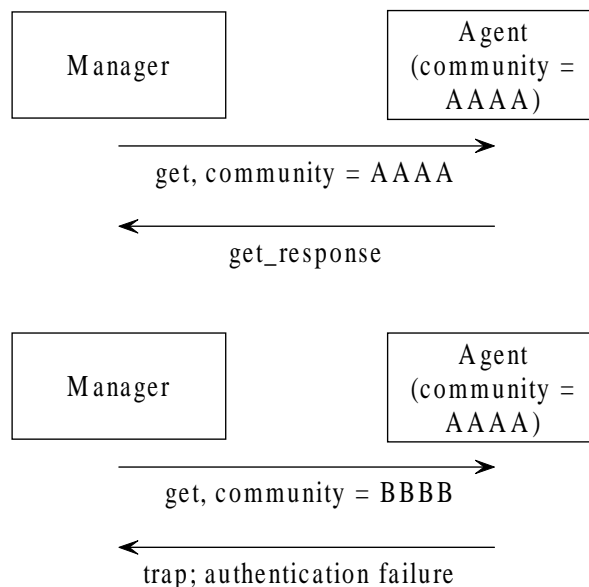
name1	value1	name2	value2	namen	valuen
-------	--------	-------	--------	-----	-----	-------	--------

variablebindings



Ver	version. SNMP version (rfc 1157 is version 1)
Comm	community. The name of the community acts as a password to authenticate the SNMP message.
request-id	Used to distinguish among outstanding requests by providing each request with a unique ID.
error-status	Used to indicate that an exception occurred while processing a request; noError (0), tooBig (1), noSuchName (2), badValue (3), readOnly (4), genErr (5).
error-index	When error-status is nonzero, may provide additional information by indicating which variable in a list caused the exception. (A variable is an instance of a managed object).
variablebindings	A list of variable names and corresponding values. (In some cases such as a get_request PDU, the values are null).
enterprise	Type of object generating trap; based on sysObjectID.
agent-addr	Address of object generating trap.
generic-trap	Generic type trap; e.g. ColdStart (0).
specific-trap	Specific trap code.
time-stamp	Time since last (re)initialisation of the network entity and the generation of the trap; contains the value of sysUpTime.

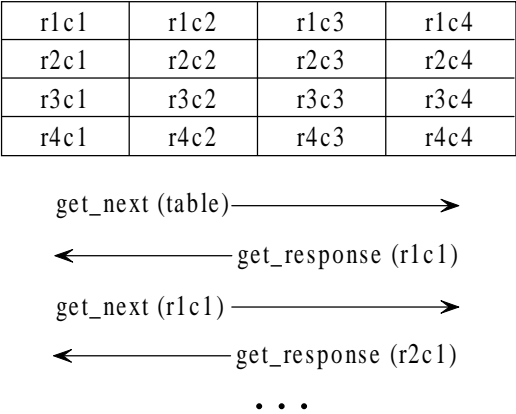
SNMP Security



Community Name – defines an access environment for a group of NMSs. NMSs within the community are said to exist within the same administrative domain. Community names serve as a weak form of authentication because devices that do not know the proper community name are precluded from SNMP

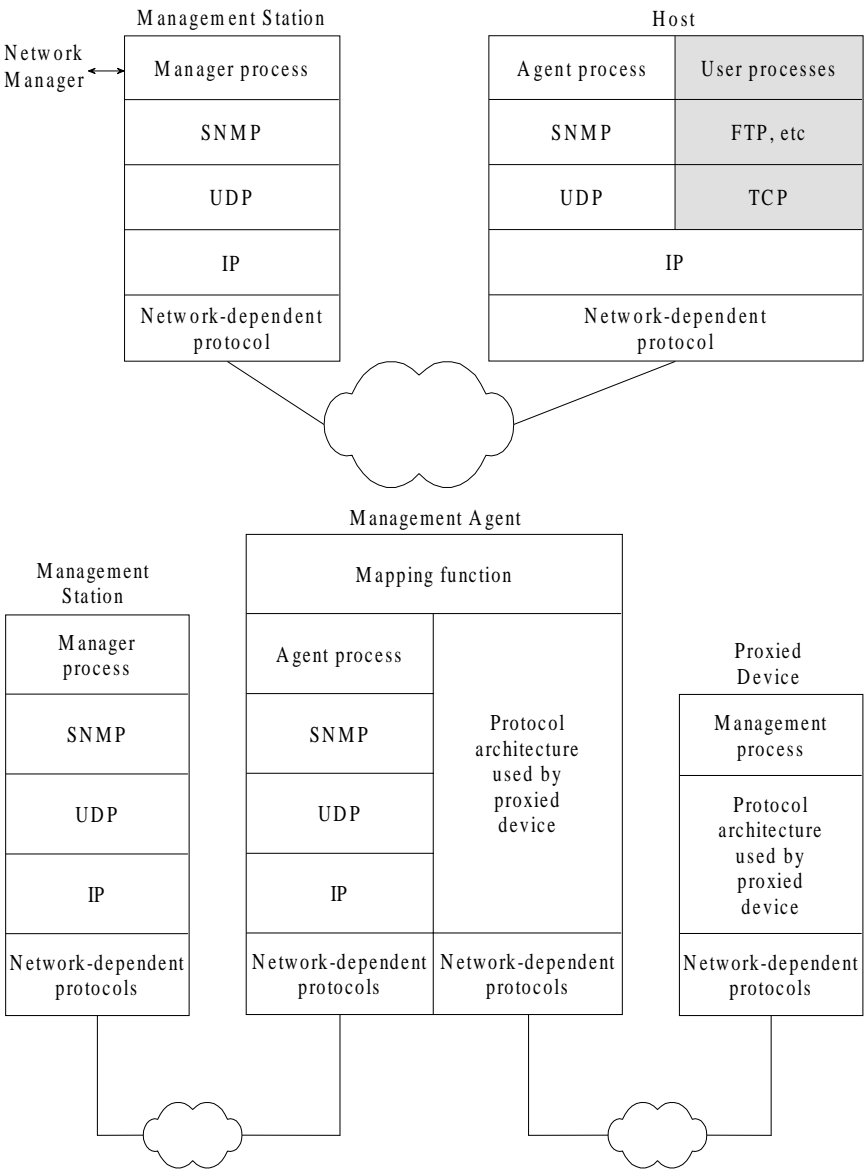
operations. The community string is transferred in a field within all SNMP data units. Agents are configured to react to one or more community strings. Some strings may give read-only access and some read-write access. The community string is transferred as plain text and is clearly a security risk.

Tables & get_next



Getting data from a table requires numerous get_next commands. get_next extracts the same piece of data about the next item e.g. a series of get_nexts can provide a list of IP addresses for different interfaces in a router.

SNMP Architecture & SNMP Proxies



To manage a device not running TCP/IP a proxy agent is required. The management station sends queries about the device to its proxy agent which converts each query into the management protocol used by the device. The reply received by the agent is then passed back to the management station. Event notification from the device is transmitted to the proxy and this is passed on to the management station in the form of a trap message.

SNMPv2

New message types:

- InformRequest: manager-manager communication allows 'peer processes' to co-operate in management.
- GetBulkRequest: retrieves more than one data item. Used instead of multiple get_next_requests.

CMIS/CMIP

Common management information services/protocol. Originally intended to be run on a fully implemented OSI protocol stack.

Very different from SNMP/SNMPv2.

- SNMP — simple agent — workload done by manager.
- CMIS — agent and manager are peer open systems; the management workload is shared more equally.
- CMIS contains:
 - Association — establish links between peers.
 - Notification — event reports (traps).
 - Operation — get/set data plus remote action and object instance creation.
- CMIP offers confirmed or unconfirmed services.

CMOT — Common Management Information Protocol/Services over TCP/IP

The *Common Management Information Services and Protocol over TCP/IP (CMOT)* proposes to implement the CMIS services on top of the TCP/IP suite as an interim solution until extensive deployment of the OSI protocol stack solution.

Leinwand, A. & Conroy, K. F. (1996, p. 190)

Comparison between CMIP and SNMP in a Fault Situation

SNMP fault situation:

- agent sends trap command to manager to report occurrence of an extraordinary event;
- manager proceeds to poll the agent and to pull back information in a series of polls to build up picture of the situation.

CMIP fault situation:

- agent compiles a comprehensive fault report and sends this to the manager as part of 'Event Report';
- command manager has all information needed so there is no need for polling.

Advantages of SNMP:

- TCP/IP widely implemented.
- SNMP is widely supported by network component and management vendors.
- The memory requirements are minimal making it easier to establish in managed devices.
- SNMP is suited to smaller networks with frequent changes in network status.